AMENDMENTS TO THE SPECIFICATION:

Please replace the paragraph [002] on page 1 in the specification with the following replacement paragraph:

— This patent application claims the benefit of priority of U.S. Provisional Application No. 60/429,371, filed November 27, 2002, which is incorporated herein by reference. This patent application is related to a series of other patent applications simultaneously filed with the present application on November 26, 2003. Those other patent applications include U.S. Patent Application Serial No. 10/721,869 entitled "DATA ELEMENT AND STRUCTURE FOR DATA PROCESSING," U.S. Patent Application Serial No. 10/721,898 entitled "COMPUTERIZED REPLICATION OF DATA OBJECTS," and U.S. Patent Application Serial No. 10/721,426 entitled "METHOD AND SOFTWARE APPLICATION FOR AVOIDING DATA LOSS." This patent application and the noted other patent applications have common inventors and are assigned to a common entity. —

Please replace the paragraph [009] on page 4 in the specification with the following replacement paragraph:

— The foregoing background and summary are not intended to be comprehensive, but instead serve to help artisans of ordinary skill understand the following implementations consistent with the invention set forth in the appended claims. ~~In addition, the foregoing background and summary are~~ —

Please replace the paragraph [017] on page 5 in the specification with the following replacement paragraph:

— Fig. 7 is a flow diagram of a second implementation of a data replicating process using the electronic data element consistent with the present invention. —

Please replace the paragraph [066] spanning pages 16-17 in the specification with the following replacement paragraph:

— Fig. 1 is a schematic block diagram for illustrating an implementation of an electronic data element within a computer system consistent with the present invention. Fig. 1 shows a computer system 101 comprising a computer 103 having a CPU 105, a working storage 102 (memory), in which ~~an~~ a software application 111 is stored for being processed by CPU 105. Software application 111 comprises program modules 109, 110 for carrying out data replication and data processing consistent with the present invention. The electronic data elements may be implemented in a table 106 comprising a column for identifiers (abbreviated as "ID" and numbered consecutively, e.g. ID1, ID2, ID3, the latter being defined as default ID, abbreviated as "DID"), a column for the state and a column for the default quality. Table 106 may be stored in memory 102. The default quality may be implemented by adding – in general sense - "yes" in a field for the default status ("default?" in Fig. 1)). Computer System 101 may further comprise input means 113~~,~~ coupled to output means 112 for interaction with a user, e.g. for starting the program modules and/or for data input, and general input/output means 104, including a ~~net~~ network connection ~~114,~~ for sending and receiving data. A plurality of computer systems 101 may be connected via the ~~net~~ network connection ~~114~~ 104 in the form of a network ~~113~~ 114. In this case the network computers ~~113~~ 114 may be used as further input/output means, including the use as further storage locations. Computer system 103 may further comprise a first storage means ~~107~~108, for example a magnetic or optical disk drive, in which the data objects of the source system may be stored. A second storage means ~~108~~107, for example a magnetic or optical disk drive, may be the storage means of the target system. Within the terms of this description, the source system in Fig. 1 is the computer system 103 minus storage means 107, and the target system is the computer system 103 minus the storage means 108. However, the target or source system may be any other network computer 114. —

Please replace the paragraph [074] spanning pages 20-21 in the specification with the following replacement paragraph:

— Fig. 4 is a flow diagram of a second implementation of a data processing operation using the electronic data element using default IDs consistent with the present invention. In a stage 401 a software application or program module (abbreviated as "SA") processes a data object in a computer system. The SA reads in a stage 402 from a table 409 the actual DID in the computer system. The SA then tries to set in stage 403 a shared lock on said DID. In stage 404 it is checked, whether the state of the DID is I or II. If no, i.e. if the state is III, the SA returns to stage 402. This loop may be run through a predefinable number of times and may, in case no lock could be set, then be broken off with an error message. The setting of the shared lock in 403 and the check whether it is successfully set may be implemented as one "atomic" stage. This means that both stages are executed essentially at the same time or, in other words, the time gap between both stages is essentially zero. If the state is I or II, the SA assigns the DID to the processed data object in stage 405 and stores the data object in stage 406. After the commit of the storing in stage 407 the shared lock set in 403 is deleted in stage 408. In the stage 407 ~~in~~ the SA may be included in a loop that ~~the SA~~ waits for the commit of the storing process from the system routine, which writes the data object to the storage device. Again, this loop may be run through a predefinable number of times and may, in case of no commit, then be broken off with an error message. —

Please replace the paragraph [080] spanning pages 22-23 in the specification with the following replacement paragraph:

— Fig. 7 is a flow diagram of a second implementation of a data replicating process using the electronic data element consistent with the present invention. In the process of Fig. 7, a software application (SA) 701, which may run in parallel to processes for processing data objects, starts in a state at which an ID (the expression "electronic data element" and "ID" are used synonymously) "old" is in a state I and defined as default ID according to table 702a in order to replicate one or more data objects. ~~Table~~ Each table 702a-f comprises three columns. The first contains in its field information on the ID, i.e. the ID, the second on the state, i.e. "I" or "II" or "III", the third on the default property, i.e. "yes" or "no". In 702a the table contains one electronic data element: consisting of the fields of the table containing "old", "I", "yes". SA creates in stage 703 a new ID "new" and sets it to state I, table 702b. In stage 704, the new ID is defined as default, what means that "yes" is entered in the corresponding field of the default? column. At the same time, the old ID is redefined as non default by entering "no" into the corresponding default? field, 702c. In stage 705, the state of the old ID is changed from I to II. In stage 706, SA waits for the commit of the change in state from state I to II. SA tries to change the state of the old ID to state III, in stage 707. In stage 708, SA determines whether the change to state III was successful. If not, the process returns to state 707 and continues to set the old ID in table 702e to state II. On the other hand, if SA successfully changes the state of the old ID at stage 707, then the old ID is set to state III in table 702f. After the old ID is set to state III, in stage 709 data replicating is performed to replicate data objects assigned to the old ID from a source system to a target system. The process ends at stage 710. —

Please replace the paragraph [085] on page 25 in the specification with the following replacement paragraph:

— The use of the inventive method ~~end~~ and electronic data element, as described in the preceding sections, ~~assures,~~ assures that no data object can be overlooked by replicating applications. —